Spatial Vision, Vol. 13, No. 2,3, pp. 287–296 (2000)

Learning in brains and machines

TOMASO POGGIO and CHRISTIAN R. SHELTON

Center for Biological and Computational Learning, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Received 7 June 1999; accepted 15 February 2000

Abstract—The problem of learning is arguably at the very core of the problem of intelligence, both biological and artificial. In this paper we sketch some of our work over the last ten years in the area of supervised learning, focusing on three interlinked directions of research: theory, engineering applications (that is, making intelligent software) and neuroscience (that is, understanding the brain's mechanisms of learning).

1. INTRODUCTION

Learning is now perceived as the gateway to understanding the problem of intelligence. Since seeing is intelligence, learning is also becoming a key to the study of artificial and biological vision. In the last few years both computer vision (which attempts to build machines that see) and visual neuroscience (which aims to understand how our visual system works) are undergoing a fundamental change in their approaches. Visual neuroscience is beginning to focus on the mechanisms which allow the cortex to adapt its circuitry for learn a new task. Instead of building a hardwired machine or program to solve a specific visual task, computer vision is trying to develop systems that can be trained with examples of any of a number of visual tasks. Vision systems that *learn and adapt* represent one of the most important directions in computer vision research. This reflects an overall trend: to make intelligent systems that do not need to be fully and painfully programmed. It may be the only way to develop vision systems that are robust and easy to use in many different tasks. As a consequence of this new interest in learning, we are witnessing a renaissance of statistics and function approximation techniques and their applications to domains such as computer vision. In this paper we sketch some of our work over the last ten years in the area of supervised learning, focusing on three interlinked directions of research: theory, engineering applications (making intelligent software), and neuroscience (understanding the brain's mechanisms of learning).

2. LEARNING AND REGULARIZATION

We have mainly concentrated on one aspect of learning: *supervised learning*. Supervised learning, or *learning-from-examples*, refers to systems that are trained, instead of programmed, by a set of examples, that is input-output pairs (x_i, y_i). At run-time they will hopefully provide a correct output for a new input not contained in the training set. One way to set the problem of learning-from-examples in a mathematically well-founded framework is the following. Supervised learning can be regarded as the regression problem of interpolating or approximating a multivariate function from sparse data. The data are the examples. Generalization means estimating the value of the function for points in the input space in which data are not available.

Once the ill-posed problem of learning-from-examples has been formulated as a problem of function approximation, an obvious approach to solving it is *regularization*. Regularization imposes a constraint of smoothness on the space of approximating functions by minimizing an appropriate cost functional.

A key result is that, under rather general conditions, the solution of the regularization formulation of the approximation problem can be expressed as the linear combination of basis functions, centered on the data points and depending on the input x. The form of the basis function K depends on the specific smoothness prior. As observed by Poggio and Girosi (1990) (and for the special case of Radial Basis Functions by Broomhead and Lowe (1988)), the solution provided by regularization rewritten as a network with one hidden layer containing as many units as examples in the training set. We call these networks Regularization Networks (RN). The coefficients c_i that represent the 'weights' of the connections to the output are set during learning (Girosi *et al.*, 1995).

An interesting special case arises for radial K. The usual example is a Gaussian These RBF networks consist of units each tuned to one of the examples with a bell-shaped activation curve. In the limit of very small σ for the variance of the Gaussian basis functions, RBF networks become look-up tables.

We notice here that the new technique of Support Vector Machines (SVM), proposed by Vapnik (1995), is closely connected to regularization (see Girosi, 1998 and Evgeniou *et al.*, 1999).

3. OBJECT DETECTION WITH SUPPORT VECTOR MACHINES

So, one can only ask, does all of the theory mean anything? We describe a system that can be trained to classes of objects such as faces or pedestrians.

3.1. Face detection

For object detection, we seek to identify the position and scale of all of the desired objects in the image. A small sub-window of the image is shifted across the entire

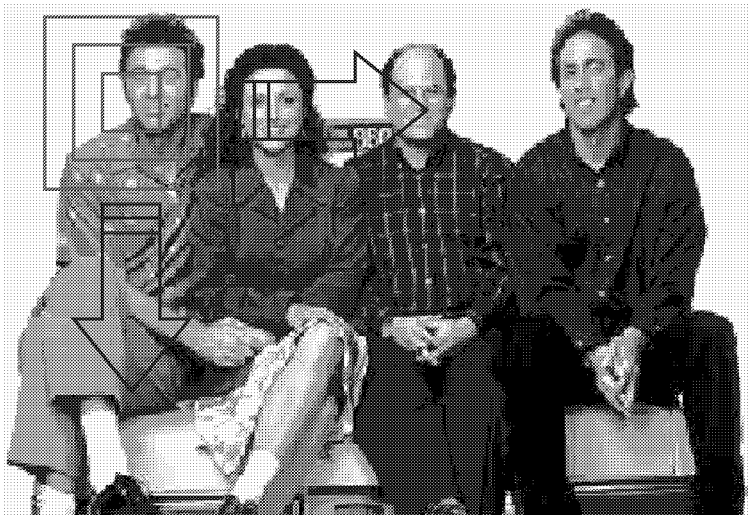

Figure 1. Sub-window scanning method to identify objects: the sub-window is translated and scaled over the entire image. At each step, the contents of the window are fed into a classifier to determine if the object exists at that location and scale.

image. At each shift, the sub-window is fed into the learned classifier to determine if the object of interest is present. To achieve multi-scale detection, we incrementally resize the image and run the detection window over each of these resized images. This scheme is shown in Fig. 1.

The first step is to build an appropriate representation of the subimage for the classifier. For this, we build an overcomplete, multiscale set of the absolute values of Haar wavelets as the basic dictionary with which to describe shape. In the case of pedestrian detection this results in roughly 1300 coefficients for each subwindow and about 400 when the system is trained to detect faces (because the subwindow is smaller). The full system is described in depth in Oren *et al.* (1997), Papageorgiou (1997), and Papageorgiou *et al.* (1998a, b).

To gauge the performance of a detection system, it is necessary to analyze the full ROC curve which gives an indication of the tradeoff between accuracy and the number of false positives gathered around MIT and over the Internet. Figure 2 compares the ROC curves of several different incarnations of our system.

From the ROC curve, it is clear that most of the impact on performance comes from what features are used; the complexity of the classifier is secondary. As expected, using color features results in a more powerful system. The curve of the system with *no feature selection* is clearly superior to all the others. This indicates that for the best accuracy, using all the features is optimal. When classifying using this full set of features, we pay for the accuracy through a slower system. Examples of processed images are shown in Fig. 3; these images were not part of the training set.

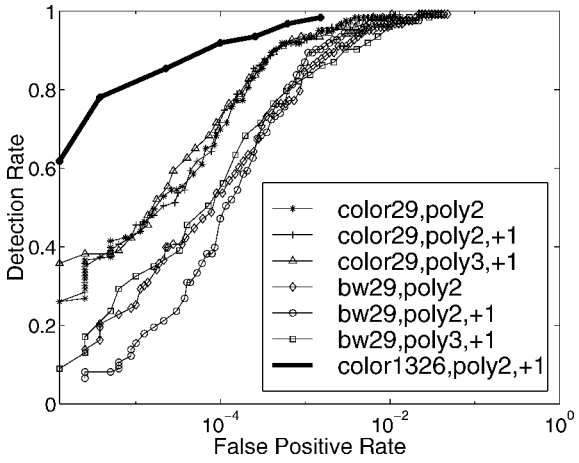

Figure 2. ROC curves for different detection systems. The detection rate is plotted against the false positive rate, measured on a logarithmic scale. The false detection rate is defined as the number of false detections per inspected window.

Figure 3. Results from our pedestrian detection system. Typically, missed pedestrians are due to occlusion or lack of contrast with the background. False positives can be eliminated with further training.

4. OBJECT RECOGNITION IN IT CORTEX

Ten years ago an example-based approach to object recognition, based on Gaussian Radial Basis Functions, suggested a view-based approach to recognition (Poggio

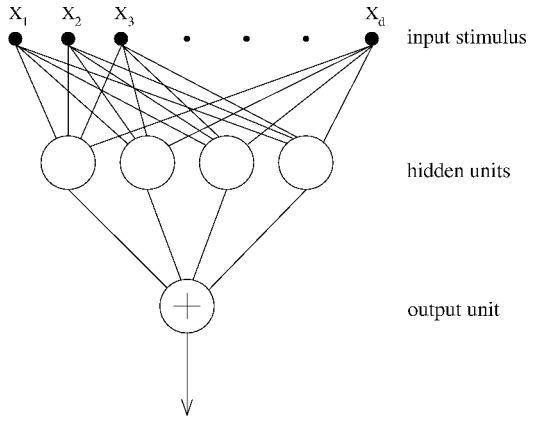

Figure 4. A Gaussian RBF network with four units which, after training, are each tuned to one of the four training views shown in the next figure. The resulting tuning curve of each of the unit is also in the next figure. The units are view-dependent and selective, relative to distractor objects of the same type.

and Edelman, 1990). Different simulations with artificial (Poggio and Edelman, 1990) and real 'wire-frame' objects (Brunelli and Poggio, 1991) and also with images of faces (Beymer, 1993; Romano, 1993) show that a view-based scheme of this type can be made to work well.

It was natural to ask whether a similar approach may be used by our brain. As Poggio and Girosi (1989) and Poggio (1990) argued, networks that learn from examples have an obvious appeal from the point of view of neural mechanisms and available neural data. Over the last ten years many psychophysical experiments (for the first such work see Bülthoff and Edelman, 1992) have supported the example-based and view-based schemes that we suggested as one of the mechanisms for object recognition. Recent physiological experiments have provided a suggestive glimpse on how neurons in IT cortex may represent objects for recognition. The experimental results seem to agree to a surprising extent with the model (Logothetis *et al*., 1995). Even more recently, we have developed a more detailed model of the circuitry and the mechanisms underlying the properties of the view-tuned units of the model (Riesenhuber and Poggio, 1998).

Figure 4 shows our basic module for object recognition. Classification of a visual stimulus is accomplished by a network of units. Each unit is broadly tuned to a particular view of the object. We refer to this optimal view as the center of the unit and to the unit as a *view-tuned unit.* One can think of it as a template to which the input is compared. The unit is maximally excited when the stimulus exactly matches its template but also responds proportionately less to similar stimuli. The weighted sum of activities of all the units represents the output of the network. The simplest recognition scheme of this type is the Gaussian RBF network: each center stores a sample view of the object and acts as a unit with a Gaussian-like recognition field around that view. At the output of the network the activities of the various units

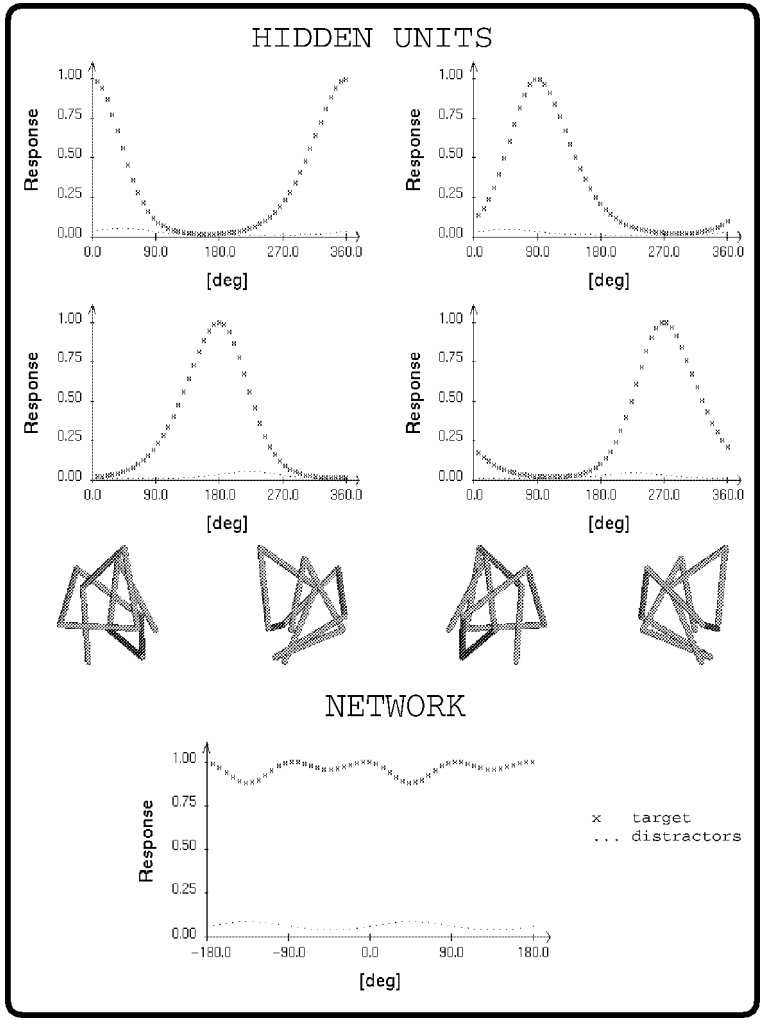

Figure 5. Tuning of each of the four hidden units of the network of the previous figure for images of the 'correct' 3D objects. The tuning is broad and selective: the dotted lines indicate the average response to 300 distractor objects of the same type. The bottom graphs show the tuning of the output of the network after learning (that is computation of the weights c): it is view-invariant and object specific. Again the dotted curve indicates the average response of the network to the same 300 distractors. From Vetter and Poggio, unpublished.

are combined with appropriate weights, found during the learning stage, in a view-invariant cell.

This example is clearly a caricature of a view-based recognition module but it helps making the main points of the argument. Physiological experiments in Logothetis' lab confirmed the main predictions of the model and found additional information about properties of IT cells (Logothetis *et al.*, 1995; Logothetis and Pauls, 1995). Supporting the model, Logothetis and coworkers found a significant

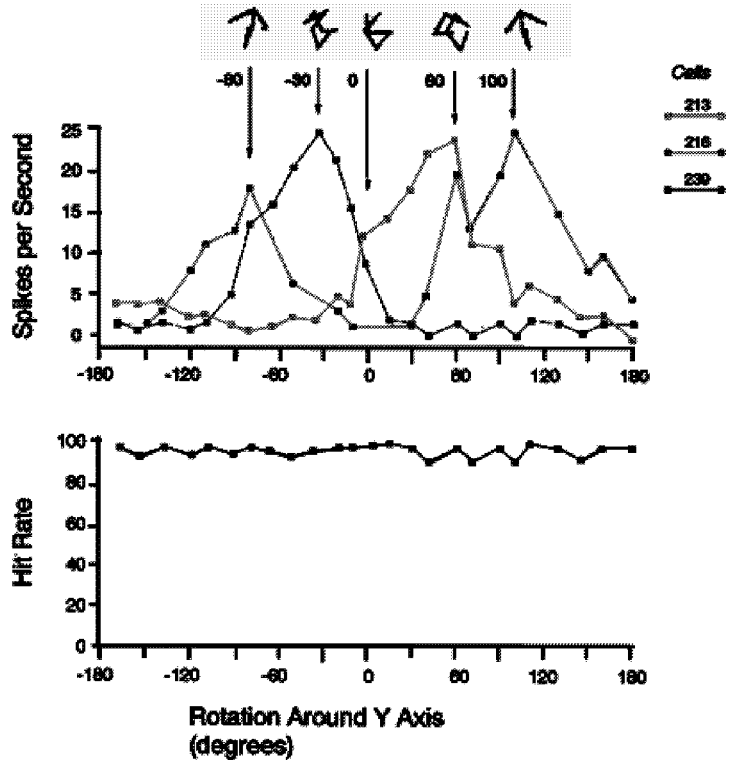

Figure 6. The top graph shows the activity of three units in IT cortex, as a function of the angle of the stimulus view. The three neurons are tuned to three different views of the same object. One of the units shows two peaks for two mirror symmetric views. The neurons firing rate was significantly lower for all distractors (not shown here). The bottom graph represents the almost perfect, view-invariant behavioral performance of the monkey for this particular object to which he was extensively trained (from Logothetis and Pauls, unpublished, 1995).

number of units that showed a remarkable selectivity for individual views of wire objects that the monkey was trained to recognize.

Figure 6 shows the responses of three units that were found to respond selectively to four different views of an wire object (Wire 71). The animal had been exposed repeatedly to this object, and its psychophysical performance remains above 95% for all tested views, as can be seen in the lower plot of Fig. 6. The figure is surprisingly similar to Fig. 5 showing the response of the view-tuned hidden units of the model of Fig. 4.

The main finding of this study is that there are neurons in IT cortex with properties intriguingly similar to the 'cartoon' model of Fig. 4. Several neurons showed a remarkable selectivity for specific views of a computer-rendered object that the monkey had learned to recognize. A much smaller number of neurons were object-specific but view-invariant, as expected in a network in which 'complex'-like view-invariant cells are fed by view-centered 'simple'-like units.

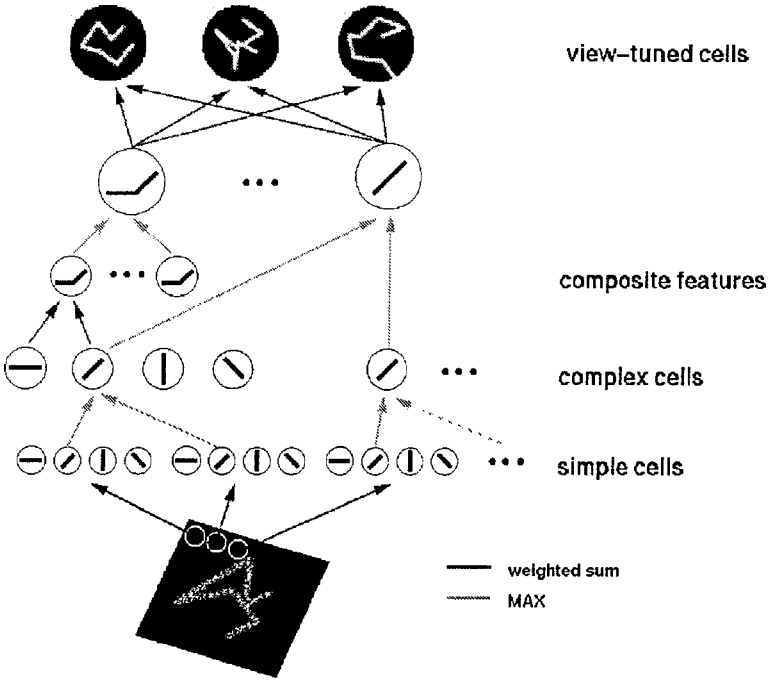

Figure 7. Model to explain receptive field properties of the view-tuned units of Fig. 4 found in experiments (from Riesenhuber and Poggio, in preparation).

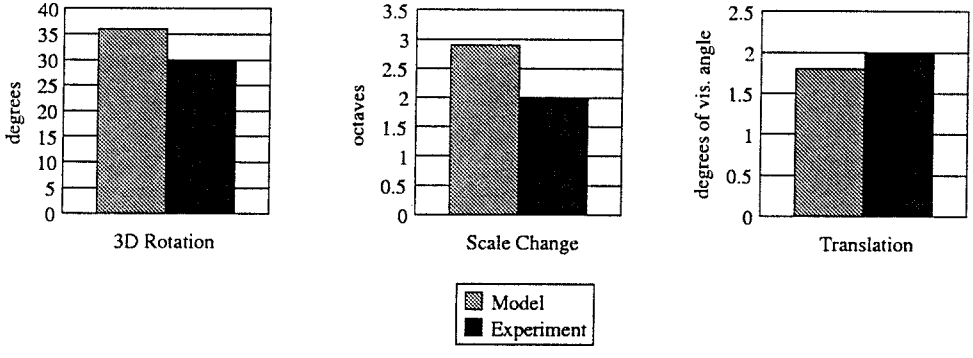

Figure 8. Comparison of theoretical model and experimental data.

More recently we have tried to address the question of the circuitry underlying the properties of the view-tuned cells. The key problem is to explain in terms of biologically plausible mechanisms their viewpoint invariance obtained from just one object view, which arises from a combination of selectivity to a specific object and tolerance to viewpoint changes. Riesenhuber and Poggio (1998) have described a model that conforms to the main anatomical and physiological constraints, reproduces all the data obtained by Logothetis *et al.* (1995) and makes several predictions for experiments on a subpopulation of IT cells. A key component

of the model is a cortical mechanism that can be used to either provide the sum of several afferents to a cell or to enable only the strongest one. The model explains the receptive field properties found in the experiment based on a simple hierarchical feedforward model. The structure of the model reflects the idea that invariance and specificity must be built up through separate mechanisms. Figure 4 shows connections to 'invariance' units in green and to 'specificity' units in blue.

A crucial element of the model is the mechanism an intermediate neuron uses to pool the activities of its afferents. Let us consider two alternative pooling mechanisms: linear summation (a special case of the weighted sum described above used to increase feature complexity) and a highly nonlinear maximum (MAX) operation (which can also be regarded as a Nearest Neighbor classification scheme), where the strongest afferent determines the response of the postsynaptic unit. It turns out that a sensible way to pool responses to achieve invariance is via a nonlinear MAX function, whereas the pooling underlying specificity is closer to a weighted sum. Simulations show agreement of the resulting model of a view-tuned cell with several physiological experiments from different labs. In particular, Fig. 4 shows the predictions of the model in comparison with experimental data.

REFERENCES

Beymer, D. J. (1993). Face recognition under varying pose. A.I. Memo No. 1461, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Broomhead, D. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks, *Complex Systems* **2**, 321–355.

Brunelli, R. and Poggio, T. (1991). Hyberbf networks for real object recognition, in: *Proceedings IJCAI*, Sydney, Australia.

Bülthoff, H. H. and Edelman, S. (1992). Psychophysical support for a 2-D view interpolation theory of object recognition, *Proc. Natl Acad. Sci. USA* **89**, 60–64.

Evgeniou, T., Pontil, M. and Poggio, T. (1999). A review of a unified framework for regularization networks and support vector machines. A.I. Memo, MIT Artificial Intelligence Lab, March.

Girosi, F., Jones, M. and Poggio, T. (1995). Regularization theory and neural networks architectures, *Neural Computation* **7**, 219–269.

Girosi, F. (1998). An equivalence between sparse approximation and Support Vector Machines, *Neural Computation* **10** (6), 1455–1480.

Logothetis, N., Pauls, J. and Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys, *Current Biology* **5** (5), 552–563.

Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constructive Approximation* **2**, 11–22.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E. and Poggio, T. (1997). Pedestrian detection using wavelet templates, in: *Proc. Computer Vision and Pattern Recognition*, Puerto Rico, pp. 193–199.

Papageorgiou, C. (1997). Object and pattern detection in video sequences, Master's thesis, MIT.

Papageorgiou, C., Evgeniou, T. and Poggio, T. (1998). A trainable pedestrian detection system, in: *Proceedings of Intelligent Vehicles*, Stuttgart, Germany, pp. 241–246.

Papageorgiou, C., Oren, M. and Poggio, T. (1998). A general framework for object detection, in: *Proceedings of the International Conference on Computer Vision*, Bombay, India.

Poggio, T. (1990). 3D object recognition: on a result by Basri and Ullman, Technical Report # 9005–03, IRST, Povo, Italy.

Poggio, T. and Edelman, S. (1990). A network that learns to recognize 3D objects, *Nature* **343**, 263–266.

Poggio, T. and Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks, *Science* **247**, 978–982.

Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. in: *Algorithms for Approximation*, Mason, J. C. and Cox, M. G. (Eds). Clarendon Press, Oxford.

Riesenhuber, M. and Poggio, T. (1998). Modeling invariances in inferotemporal cell tuning. A.I. Memo No. 1629, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Romano, R. (1993). Real-time face verification, Master's thesis, Massachusetts Institute of Technology.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.